Prepared by Mohammad Umer

Python Tutorial

1.1 - Print

```
In [3]:  print("hello World")              # String Literal : Sequence of characters enclosed in dou

         hello World
```

Q1.Print a poem Twinkle, twinkle, little star in python.

```
In [9]:  print("Twinkle, twinkle, little star,
         How I wonder what you are!
         Up above the world so high,                    ## SyntaxError
         Like a diamond in the sky.")
```

```
         Cell In[9], line 3
           print("Twinkle, twinkle, little star,
                 ^
         SyntaxError: unterminated string literal (detected at line 3)
```

Below is the Correct Way:

```
In [10]:  print('''Twinkle, twinkle, little star,
          How I wonder what you are!
          Up above the world so high,
          Like a diamond in the sky.''')

          Twinkle, twinkle, little star,
          How I wonder what you are!
          Up above the world so high,
          Like a diamond in the sky.
```
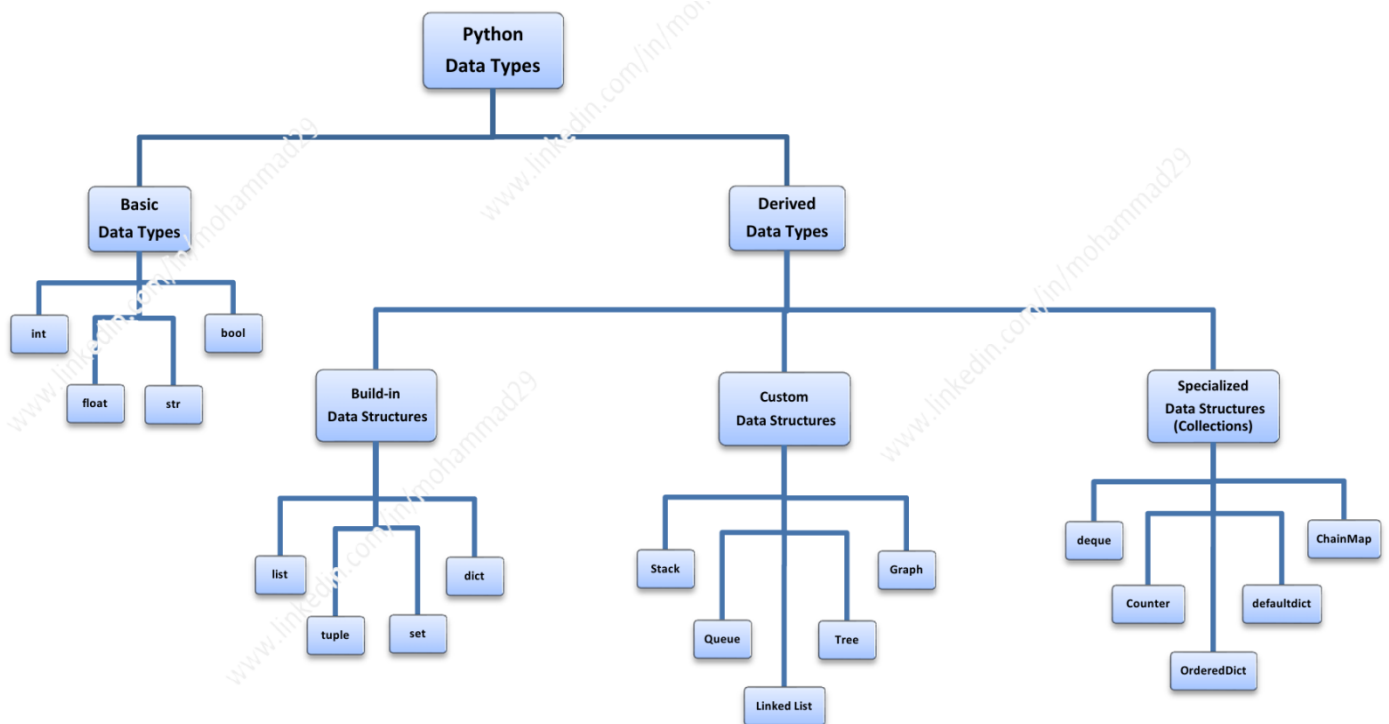
1.2 - DataTypes

Datatypes means what type of data that can be represented and manipulated.
OR
Datatypes means what type of data the variable is holding.

```
In [11]: print(type("Hello World"))          # type checking
<class 'str'>
```

1.3 - Basic Data Types

These are the fundamental data types in Python, which define the type of data that can be represented and manipulated. These are simple, primitive types that are directly supported by the language and do not require any additional libraries or modules.
It includes:

- `int:` Integer numbers (e.g., 5, -23, 100)

- `float:` Floating-point numbers (e.g., 3.14, -0.001)

- `str:` String (e.g., "hello", 'Python')

- `bool:` Boolean values (True, False)

-Integers

```
In [15]: print(29)
         print(type(29))

         29
         <class 'int'>

In [2]:  print(isinstance(29, int))

         True
```

-float

```
In [1]:  print(2.5)
         print(type(2.5))

         2.5
         <class 'float'>
```

-Strings

```
In [23]: print("29")
         print(type("29"))

         29
         <class 'str'>
```

1.3 - Type Conversion

```
In [18]:   # Converting float to integer

           print(2.5)
           print(type( int(2.5) ))

2.5
<class 'int'>
```

1.4 - Variables

Variables are fundamental elements in programming that are reversed memory locations to store data that can be referenced and manipulated in a program. In Python, `variables are created when you assign a value to them`, and they do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable.

And the name given to these variables that helps us to differentiate one variable from the another is called `Identifiers`.

- Variable Naming Conventions

- variable name cant be a keyword.
- Variable names should be descriptive
- They must start with a letter or an '_' and contains letter,numbers and underscores.
- variables names case sensitive.

- Valid variable names

first_name = "Mohammad"
last_name = "Umer"

- Invalid variable names

2age = 30
first-name = "Mohammad"
@name = "Umer"

Example 1

```
In [25]: x = 29
         print("Identity : ",id(x))                    # id is always Unique & constant of the ob
         print("Memory Location of the variable : ",hex(id(x)))
```

```
Identity :  140704678739256
Memory Location of the variable :  0x7ff85c64fd38
```

Example 2

```
In [ ]: # STRING Variable
        name = "Umer"
        print(name)
        print(type(name))
```

```
Umer
<class 'str'>
```

```
In [2]: ## Invalid
        @name = "Umer"
        print(@name)                # SyntaxError
```

```
  Cell In[2], line 2
    @name = "Umer"
    ^
SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?
```

Example 3

```
In [21]: # FLOAT
         y = 2.5
         print(y)
```

```
2.5
```

```
In [22]: # TYPE CONVERSION
         y = int(2.5)
         print(type(y))
```

```
<class 'int'>
```

Example 4

```
In [3]: #bool
        is_student = True
        print(type(is_student))
```

```
<class 'bool'>
```

- Random Value in a Variable

```
In [4]: import random
        x = random.randint(1, 100)          # Generates a random number between 1 and 100
        print(x)

        38
```

- Python is `Case Sensitive`

```
In [1]: name = "Umer"
        name = "Ahmed"

        print(name)

        Ahmed
```

```
In [2]: # Case Sensitive
        name = "Umer"
        Name = "Ahmed"

        print(name)
        print(Name)

        Umer
        Ahmed
```

1.5 - Keywords

Keywords are the `reserved words in Python` and can not be used as an variable name (identifiers).

```python
In [10]: import keyword
         print(keyword.kwlist)

['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'cont
inue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if',
'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try',
'while', 'with', 'yield']
```

```python
In [11]: len(keyword.kwlist)

Out[11]: 35
```

Example

```python
In [12]: ## Keyword cant be used as a Identifier
         try = 29                                    # SyntaxError
         print(try)
```

```
  Cell In[12], line 1
    try = 29
        ^
SyntaxError: expected ':'
```

NOTE : There is a in-build function in Python, that checks if the object is an instance of the specified type or its subclass. i.e, `isinstance(object, type)`

```python
In [5]: var = 29

        print(isinstance(var,int))

True
```